Red Hat Certificate System Enterprise Security Client Guide

Red Hat Author(s): Red Hat, Inc.

ISBN: N/A

Publication date:

Red Hat Certificate System Enterprise Security Client Guide:

Copyright © 2006 Red Hat, Inc. All rights reserved.

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0. A copy of this license is available at *http://www.opencontent.org/openpub*[1]. Any Red Hat trademarks that are incorporated in the documentation are not subject to the Open Publication License and may only be used or replicated with the express permission of Red Hat, Inc.

[1] http://www.opencontent.org/openpub

1. Introduction 1
1. Features 1
2. Platform Support 5
3. Installation 7
1. Installation on Windows 7
2. Installation on Red Hat Enterprise Linux 15
3. Installation on Mac OS X 16
4. Using the Enterprise Security Client 21
1. Launching Enterprise Security Client 21
2. Phone Home 21
3. Windows Cryptographic Service Provider 24
4. Smart Card Auto Enrollment 25
5. Customizing the Smart Card Enrollment User Interface 30
6. Managing Smart Cards 32
6.1. Formatting the Smart Card 33
6.2. Reset Smart Card Password 34
6.3. Viewing Certificates 34
6.4. Enrolling Smart Cards 35
7. Diagnosing Problems 37
5. Using Enterprise Security Client Keys for SSL Client Authentication and S/MIME 41
1. Using the Certificates on the Token for SSL 41
2. S/MIME Applications 43
6. Uninstalling Enterprise Security Client 45
1. Uninstalling on Windows 45
2. Uninstalling on Red Hat Enterprise Linux 45
3. Uninstalling on Mac OS X 45
A. Enterprise Security Client Configuration 47
1. Configuration 47
2. Enterprise Security Client Mac TokenD 48
2.1. Verifying the TokenD Is Working 49
3. Enterprise Security Client XUL and Javascript Functionality 49
3.1. Quick Javascript UI Guide 50
4. Enterprise Security Client File Locations 50
4.1. Windows 51
4.2. Red Hat Enterprise Linux 51
4.3. Mac OS X 51
Index 53

Introduction

The Red Hat Certificate System creates, manages, renews, and deletes certificates and keys within an organization. There are five subsystems which govern the behavior of the public-key infrastructure (PKI) of the organization:

- The Certificate Authority (CA), which creates, renews, and revokes certificates.
- The Data Recovery Manager (DRM), which archives and recovers keys.
- The Online Certificate Status Manager, which stores lists of revoked certificates for client applications to use to check if a certificate is valid.
- The Token Processing System (TPS), which interacts with smart cards to generate and store keys and certificates for a specific user.
- The Token Key Service (TKS), which generates and stores master keys used by the TPS.

End users can use security tokens, which are also called smart cards, to store user certificates used for applications such as single sign-on access and client authentication. End users are issued the tokens containing certificates and keys required for signing, encryption, and other cryptographic functions. To use the tokens, the TPS must be able to recognize and communicate with them. The tokens have to be *enrolled*, the process of formatting tokens with keys and certificates and adding them to the Certificate System. Enterprise Security Client provides the user interface for end entities to enroll tokens and to communicate with the TPS. Enterprise Security Client provides the conduit through which TPS communicates with each token over a secure HTTP channel (HTTPS).

After a token is enrolled, applications such as Mozilla Firefox and Thunderbird can be configured to recognize the token and use it for security operations, like client authentication and S/MIME mail. Enterprise Security Client provides the following capabilities:

- Supports Visa Open Platform-compliant smart cards like Axalto Cyberflex egate 32k tokens.
- Enrolls security tokens so they are recognized by TPS.
- Maintains the security token, such as re-enrolling a token with TPS.
- Provides information about the current status of the token or tokens being managed.
- Supports server-side key generation so that keys can be archived and recovered on a separate token if a token is lost.

1. Features

- The *Phone Home* feature defines the token issuer name, TPS server, and TPS end-entities

interface URL without requiring any user configuration.

- Enterprise Security Client has diagnostic logging that records common access and events and records potential errors such as interruptions with the connection between the Enterprise Security Client and TPS server.
- The Enterprise Security Client user interface incorporates Mozilla XULRunner technology. XULRunner is a runtime package which hosts standalone applications based on XUL, an XML markup language with a rich feature set for user interfaces. XUL has the following advantages over HTML for applications:
  - XUL provides a wide UI widget set and greater control over the presentation.
  - XUL markup is local to the client machine, so it has a greater privilege level than HTML.
  - XUL also uses Javascript as the scripting language for convenient program logic scripting.
  - XUL Javascript code can make use of the array of Mozilla functionality by using their XPCOM technology.
- The Mac Enterprise Security Client ships with a smart card-specific TokenD component which bridges the gap between Certificate System-supported tokens and the Mac CDSA security layer, allowing current OS X applications like Apple Mail and Safari to take advantage of the capabilities of Certificate System tokens:
  - The Mac Keychain Access utility can be used to view the certificates and keys on Certificate System tokens.
  - The Apple Mail client can be used to view signed and encrypted emails using Certificate System tokens.
  - The Apple Safari browser can use Certificate System tokens to log onto secure SSL web sites.
- This version of Enterprise Security Client provides tray icon functionality on all three platforms, including tool tips for errors and actions such as inserting or removing a smart card.

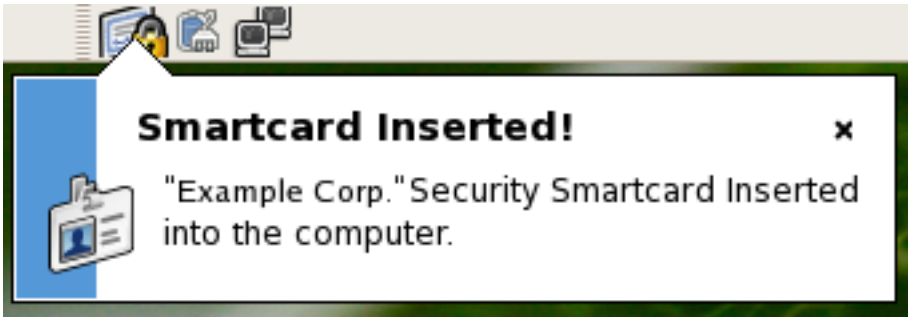


Figure 1.1. Example Token Tray Icon and Tool Tip

On most operating systems, many programs maintain an icon in the tray or notification area. These icons can be used to control the operation of the program, usually through context menus when the icon is right-clicked. In the default Enterprise Security Client configuration, Enterprise Security Client launches and automatically minimizes to the tray. This tray functionality behaves differently on the different operating systems:

- *Windows.* When right-clicked, the tray icon shows a simple menu with options to **Manage Smart Card**, which opens the Enterprise Security Client interface, and to **Exit Smart Card Manager**, which exits the Enterprise Security Client. The exit option in that menu is the only want to exist the Enterprise Security Client on Windows; clicking the **X** in the top right corner minimizes Enterprise Security Client to the tray. Double-clicking the tray icon brings Enterprise Security Client to the front. There are also notification messages, shown as standard balloon tooltips, on events like inserting or removing a card.

- *Linux.* The tray icon appears only if the notification area in Gnome has been enabled. The tray icon options are identical to the Windows options. Clicking the **X** in the top left corner closes the current window and minimizes Enterprise Security Client to the tray.

- *Mac.* On Mac, the tray is called the dock. Since Enterprise Security Client is based on Mozilla, right-clicking on the Enterprise Security Client dock icon reveals all the standard Mozilla Firefox menu options, including options to hide, show, and quit the client. The Enterprise Security Client also has a menu item called **Manage Smart Cards** in the dock menu, which opens the card management UI. The top level application menu has a menu under **Go**, **Manage Smart Card**, which also opens the card management window.

Platform Support

Enterprise Security Client supports the following platforms:

- Red Hat Enterprise Linux 4 AS (Intel x86)
- Red Hat Enterprise Linux 4 ES (Intel x86)
- Microsoft Windows XP
- Apple MAC OS X 10.4.x (Tiger)

Smart Card Support.

Enterprise Security Client supports the following smart cards:

- Visa Open Platform-compliant smart cards such as Axalto Cyberflex egate 32k tokens

Installation

Enterprise Security Client is packaged as a set of RPMs and other files that are part of the complete Certificate System distribution. These are listed in the installation chapter of the *Certificate System Administrator's Guide*.

The first step for installing Enterprise Security Client is successfully obtaining the needed Enterprise Security Client packages. The *Certificate System Administrator's Guide* explains how to retrieve these RPMs and other files through the **Red Hat Certificate System 7.2 (AS v.4 for x86)** or **Red Hat Certificate System 7.2 (ES v.4 for x86)** Red Hat Network channels. There are two ways to obtain the packages:

- Downloading an ISO image or packages through the Red Hat Network channel
- Using the Red Hat `up2date` utility

On Linux platforms, the preferred method of obtaining RPMs is using the `up2date` command-line utility.

```
up2date esc
```

If the `up2date` process is successful, all of the necessary Enterprise Security Client RPMs are installed and ready for use.

Both the Mac and Windows Enterprise Security Client bundles are available only in the **Downloads** area of the Red Hat Network. There are two channels for the packages, one for 32-bit and one for 64-bit; Mac and Windows clients are only available in 32-bit.

- The Mac Enterprise Security Client package is `ESC.dmg`.
- The Windows Enterprise Security Client package is `Smart Card Manager Setup.exe`.

1. Installation on Windows

To install Enterprise Security Client on Windows, do the following:

1. Obtain the Windows Enterprise Security Client installer `Smart Card Manager Setup.exe` from the Red Hat Network channel.
2. Double-click the `Smart Card Manager Setup.exe` file to launch the Enterprise Security Client installation program.

Setup - Smart Card Manager
Welcome to the Smart Card Manager Setup Wizard
This will install Smart Card Manager 1.0.0-16 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

Figure 3.1. Launching the Installation Wizard

3. The wizard displays the list of packages which will be installed.

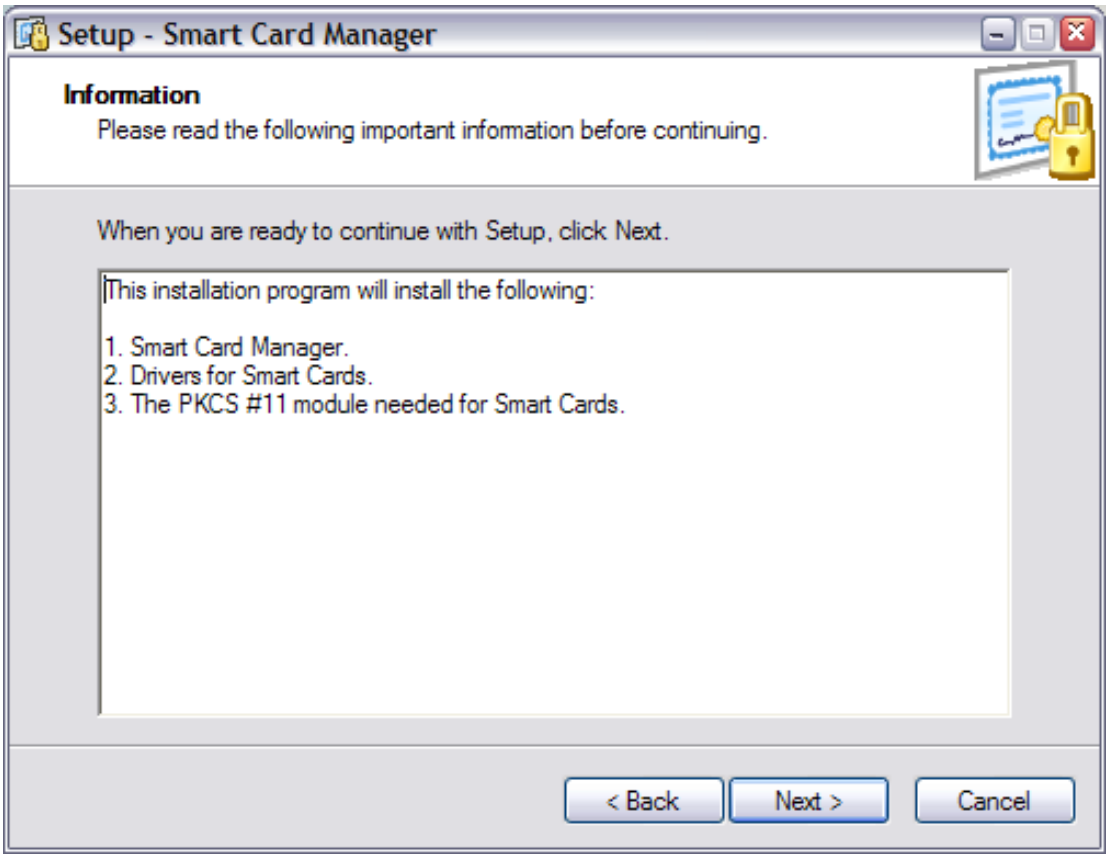

Setup - Smart Card Manager
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
This installation program will install the following:
1. Smart Card Manager.
2. Drivers for Smart Cards.
3. The PKCS #11 module needed for Smart Cards.
< Back
Next >
Cancel

Figure 3.2. Launching the Installation Wizard

4. The wizard screen asks for the final installation directory for Enterprise Security Client. The default is `C:\Program Files\Red Hat\ESC`.

Setup - Smart Card Manager
Select Destination Location
Where should Smart Card Manager be installed?
Setup will install Smart Card Manager into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\Red Hat\ESC
Browse...
At least 18.0 MB of free disk space is required.
< Back
Next >
Cancel

Figure 3.3. Installation Directory

5. The wizard screen asks for the start menu directory for Enterprise Security Client. The default is `Red Hat`.

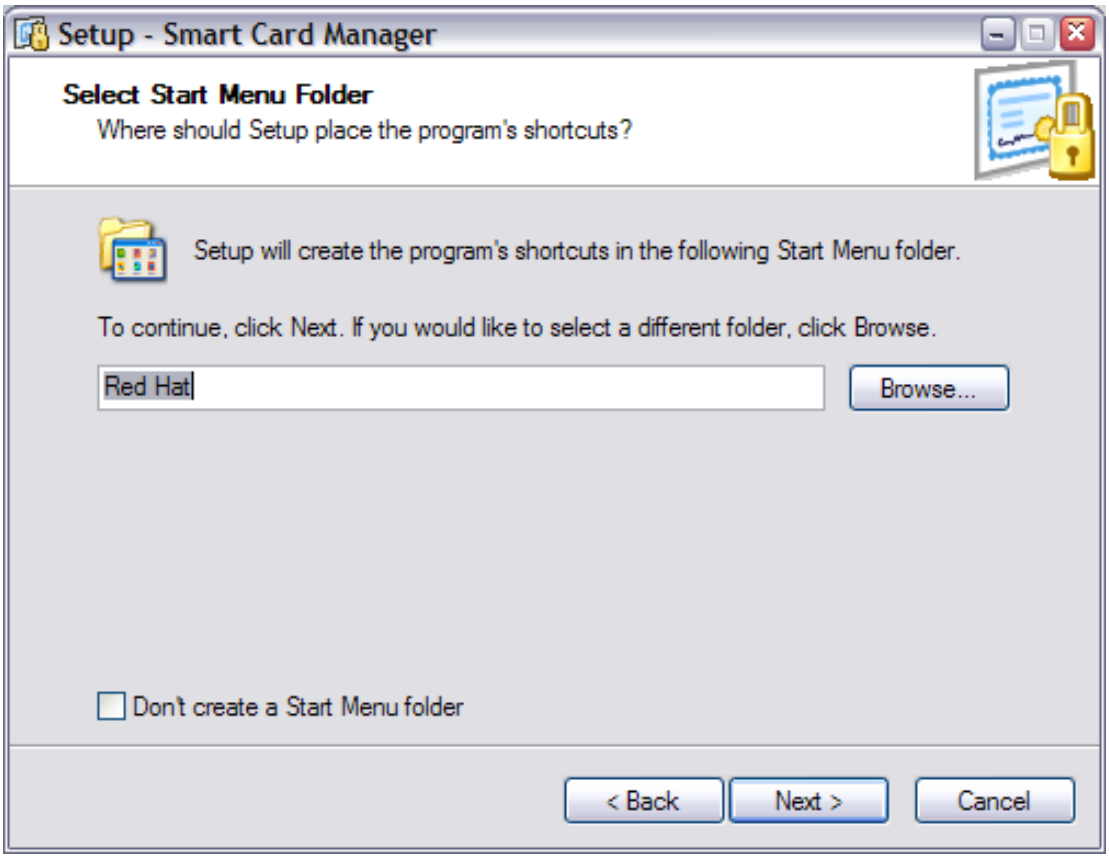

Setup - Smart Card Manager
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Red Hat
Browse...
Don't create a Start Menu folder
< Back
Next >
Cancel

Figure 3.4. Start Menu Directory

6. Proceed through the Enterprise Security Client installation wizard. Click **Install** to begin installing the Enterprise Security Client components.

NOTE

The installation process also installs the CoolKey PKCS #11 driver and Egate drivers needed for Certificate System-supported keys and automatically installs the Certificate System PKCS #11 module in any Mozilla browsers it can locate. The installer places the Certificate System Cryptographic Service Provider (CSP) on the user's system to allow users to use their smart cards with Microsoft products such as Outlook and Internet Explorer.

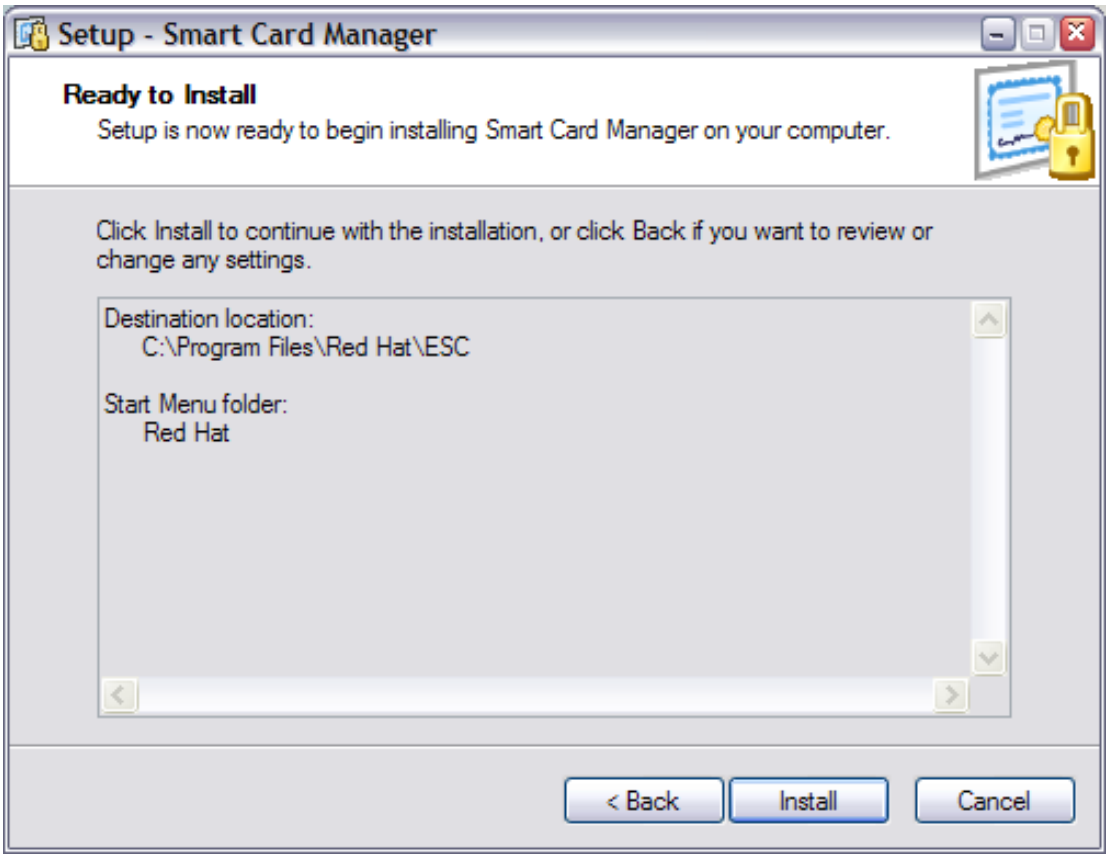

Setup - Smart Card Manager
Ready to Install
Setup is now ready to begin installing Smart Card Manager on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Program Files\Red Hat\ESC
Start Menu folder:
Red Hat
< Back
Install
Cancel

Figure 3.5. Beginning Installation

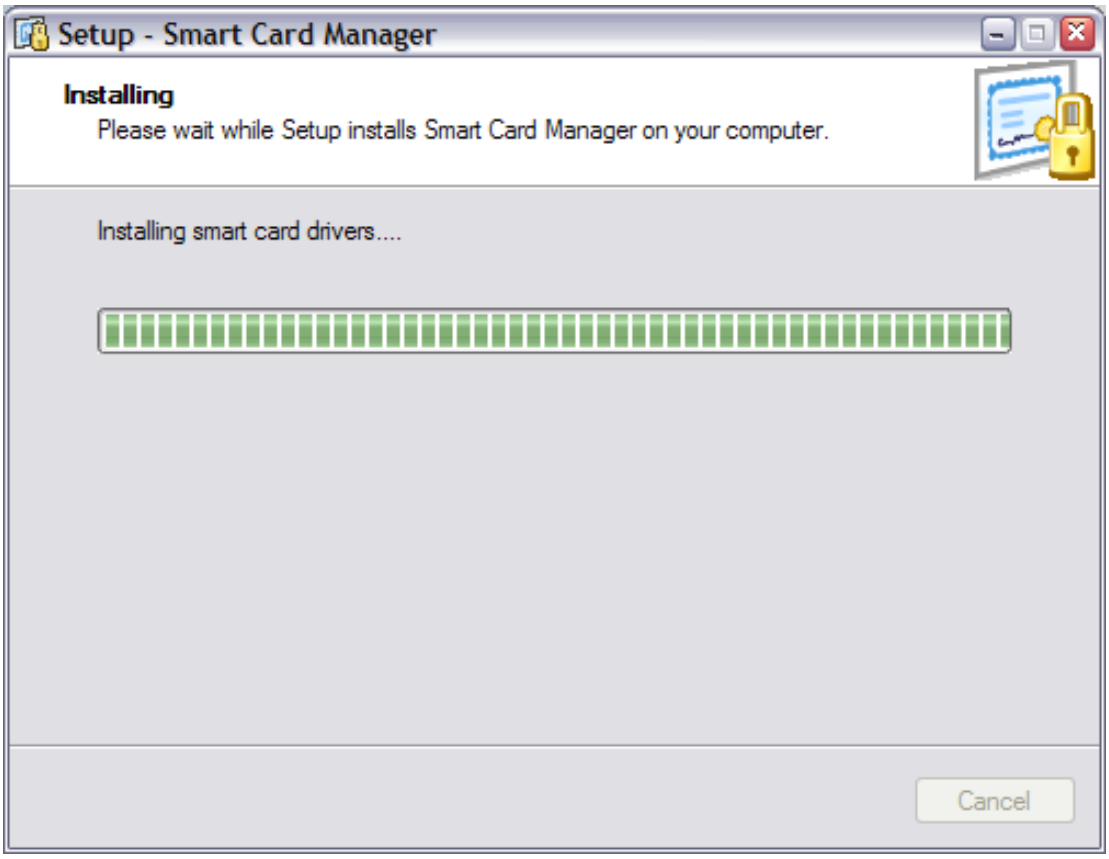

Setup - Smart Card Manager
Installing
Please wait while Setup installs Smart Card Manager on your computer.
Installing smart card drivers....
Cancel

Figure 3.6. Installation Progress

7. Once the installation has completed, Enterprise Security Client will prompt for the user to insert a token and can be launched for immediate use.

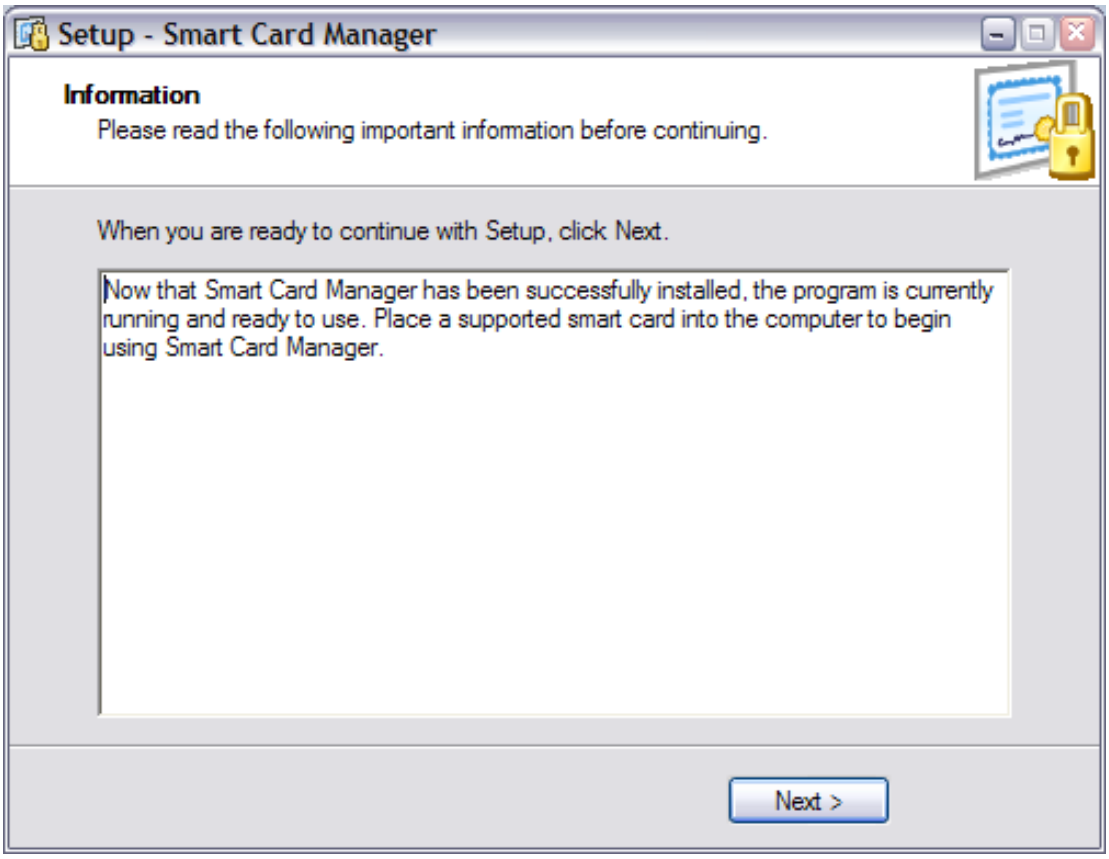

Setup - Smart Card Manager
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
Now that Smart Card Manager has been successfully installed, the program is currently running and ready to use. Place a supported smart card into the computer to begin using Smart Card Manager.
Next >

Figure 3.7. Launching the Smart Card Manager

8. Click the **Finish** button to complete the installation.

Setup - Smart Card Manager
Completing the Smart Card Manager Setup Wizard
Setup has finished installing Smart Card Manager on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
< Back
Finish

Figure 3.8. Completing Installation

2. Installation on Red Hat Enterprise Linux

To install Enterprise Security Client and its supporting components on Red Hat Enterprise Linux, do the following:

NOTE

If the `up2date` utility was already used to install Enterprise Security Client, there is no need for further installation; the client has already been installed. The following procedure is for installing from a CD-ROM image.

1. Copy the Enterprise Security Client installation RPMs packaged with Red Hat Certificate System.
2. Install the RPMs as `root` in the following order:

```
su

rpm -ivh ccid-1.0.1-5.i386.rpm
rpm -ivh pcsc-lite-1.3.1-7.i386.rpm
rpm -ivh pcsc-lite-libs-1.3.1-7.i386.rpm
rpm -ivh ifd-egate-0.05-15.i386.rpm
rmp -ivh coolkey-1.0.1-4.i386.rpm
rpm -ivh esc-1.0.0-19.i386.rpm
```

The version numbers for the different packages may be different than those listed here because of updates, patches or other releases. These are included as an example.

The Enterprise Security Client installation is located in `/usr/lib/esc-1.0.0`. The `esc` shell script is installed in `/usr/bin/esc`. Enterprise Security Client can be launched by typing `esc` at a command prompt.

Enterprise Security Client for Linux has a daemon process which runs silently, waiting for a smart card to be inserted. When an unenrolled smart card is inserted, the daemon automatically launches the client UI, and the Enterprise Security Client guides the user through the enrollment process. The client can also be launched manually by selecting **System Settings**, then **Smart Card Manager**, from the system menu.

3. Installation on Mac OS X

To install Enterprise Security Client and its supporting components on Mac OS X, do the following:

1. Obtain the `ESC.dmg` file on the Red Hat Network channel.
2. Double-click on `ESC.dmg`, exposing the **Enterprise Security Client Volume**.

   Inside the Volume are two directories, `ESC.app` and `Coolkey1.11.pkg`. `ESC.app` is the drag-able Enterprise Security Client application, and `Coolkey1.11.pkg` is the installer for the token support software, including the TokenD system.
3. Install the Enterprise Security Client. To install Enterprise Security Client, drag the `ESC.app` file to an accessible location, such as the desktop.
4. Install the CoolKey package.

   a. Double-click the `Coolkey1.11.pkg` file to launch the CoolKey installer, and follow the directions to complete installation.

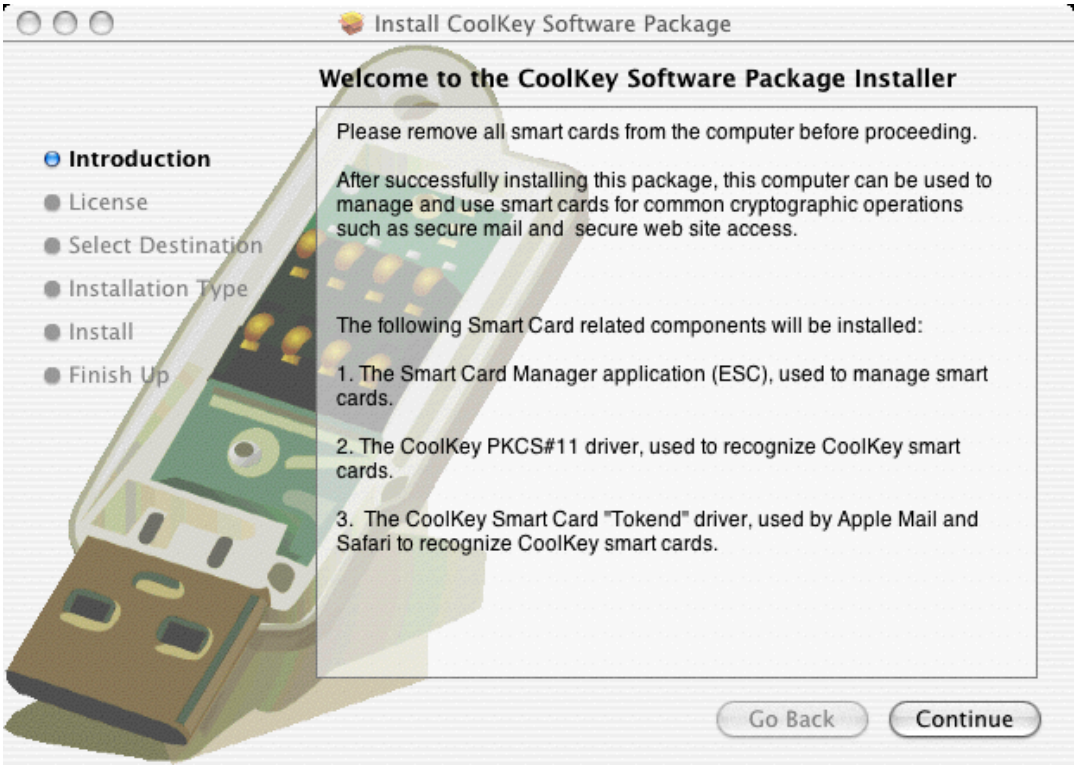

Install CoolKey Software Package
Welcome to the CoolKey Software Package Installer
Introduction
License
Select Destination
Installation Type
Install
Finish Up
Please remove all smart cards from the computer before proceeding.
After successfully installing this package, this computer can be used to manage and use smart cards for common cryptographic operations such as secure mail and secure web site access.
The following Smart Card related components will be installed:
1. The Smart Card Manager application (ESC), used to manage smart cards.
2. The CoolKey PKCS#11 driver, used to recognize CoolKey smart cards.
3. The CoolKey Smart Card "Tokend" driver, used by Apple Mail and Safari to recognize CoolKey smart cards.
Go Back
Continue

Figure 3.9. Mac Installation Program

b. Select the location to install the CoolKey package.

Figure 3.10. Installation Location

c. Click the **Upgrade** button to begin installation.

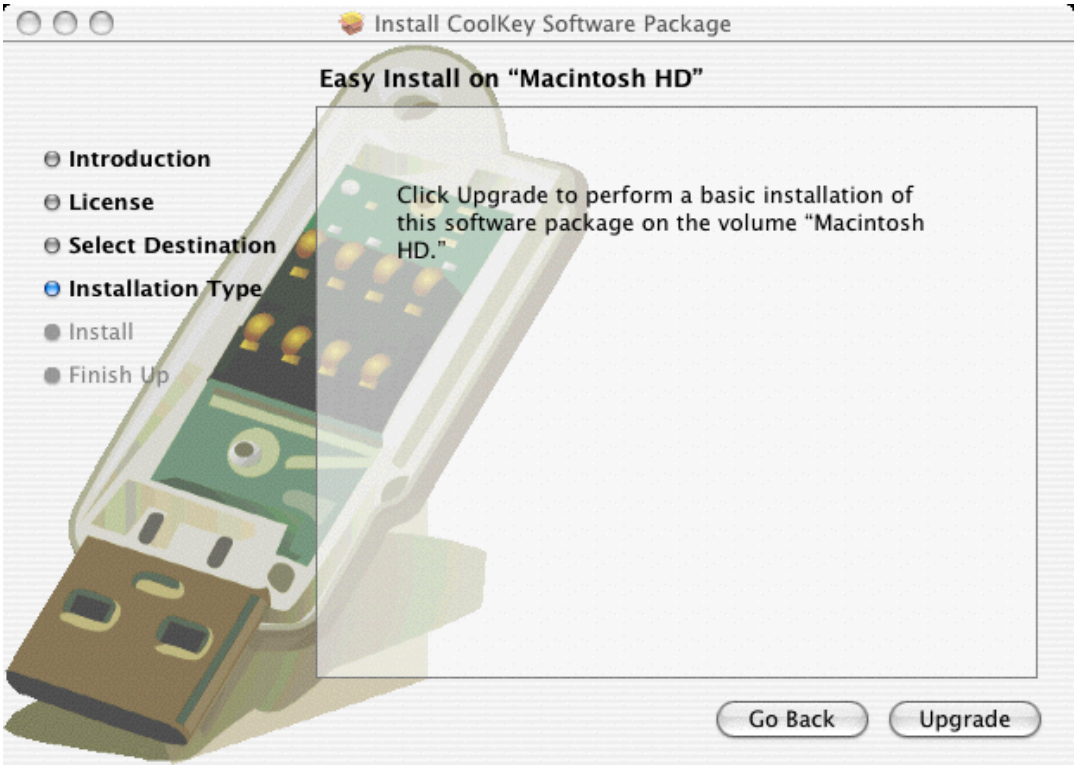


Figure 3.11. Launch Installation

d. Supply the Mac administrator password.

Figure 3.12. Mac Admin Password

e. Click the **Close** button to complete the installation.

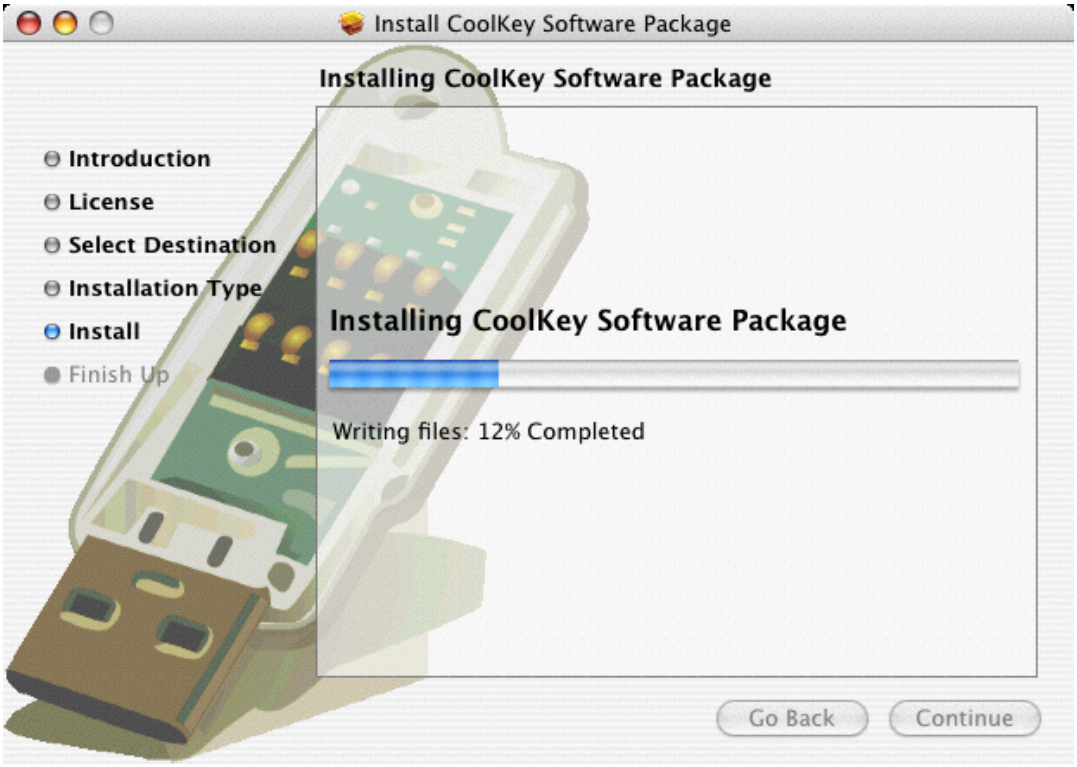


Figure 3.13. Finish Installation

When the process is completed, the Egate token drivers, the PKCS11 module, and the TokenD software are installed on the local system.

Using the Enterprise Security Client

The following section contains basic instructions on using the Enterprise Security Client for token enrollment, formating, and password reset operations.

1. Launching Enterprise Security Client

- On Red Hat Enterprise Linux 4, launch Enterprise Security Client by typing `esc` at the command prompt; this brings up the Enterprise Security Client daemon process, which silently watches for inserted smart cards. The client can also be launched by selecting **System Settings**, then **Smart Card Manager**, from the system menu.
- On Windows, Enterprise Security Client is launched from the desktop or the start menu; Enterprise Security Client is also configured to launch on reboot.
- On Mac OS X, Enterprise Security Client is launched by double-clicking the Enterprise Security Client icon wherever the client is installed.

2. Phone Home

The Enterprise Security Client offers a feature called *Phone Home* that associates information within each smart card with information which points to distinct TPS servers and Enterprise Security Client UI pages. Whenever the Enterprise Security Client accesses a new smart card, it connects to the TPS server and retrieves the Phone Home information.

Phone Home quickly retrieves and then caches this information; because the information is cached locally, the TPS subsystem does not have to be contacted each time a formatted smart card is inserted.

The information can be different for every key or token, which means different TPS servers and enrollment URLs can be configured for different corporate or customer groups. Phone Home makes it possible to configure different TPS servers for different issuers or company units, without having to configure the Enterprise Security Client manually to find the proper server and URL.

NOTE

In order for the TPS subsystem to utilize the Phone Home feature, Phone Home must be enabled in the TPS configuration file:

```
op.format.tokenKey.issuerinfo.enable=true
op.format.tokenKey.issuerinfo.value=http://server.example.com
```

Since the Enterprise Security Client is based on Mozilla XULRunner, each user has a profile similar to the user profiles used by Mozilla Firefox or Thunderbird. The Enterprise Security Client accesses the configuration preferences file. When the Enterprise Security Client caches information for each token, the information is stored in the user's configuration file. The next time the Enterprise Security Client is launched, it retrieves the information from the configuration file instead of contacting the server again.

The Phone Home information is put on the token in one of two ways:

- The preferred method is that the information is burned onto the token at the factory. When the tokens are ordered from the manufacturer, the company should also supply detailed information on how the tokens should be configured when shipped.
- If tokens are blank, the company IT department can supply the information when formating small groups of tokens.

The following information is used by the Phone Home feature for each smart card:

- The TPS server and port. For example:

```
"esc.key.40900062ff020000ba87.tps.url" =
 "http://tps.example.com:12443//nk_service"
```

- The TPS enrollment interface URL. For example:

```
"esc.key.40900062ff020000ba87.tps.url" =
 "http://tps.example.com:12443/cgi_bin/esc.cgi?"
```

- The issuing company name or ID. For example:

```
"esc.key.40900062ff020000ba87.issuer.name" = "Example Corp"
```

- The Phone Home URL. For example:

```
"esc.key.40900062ff020000ba87.phone.home.url" =
 "http://tps.example.com:12443/phone_home/phone_home.cgi?"
```

- Optionally, a default browser URL to access when an enrolled smart card is inserted.

```
"esc.key.40900062ff020000ba87.EnrolledTokenBrowserURL" =
```

```
"http://www.test.example.com"
```

The Phone Home feature and the different type of information used by it only work when the TPS has been properly configured to use Phone Home. If the TPS is not configured for Phone Home, then this feature is ignored. *Example 4.1, "TPS Phone Home Configuration File"* shows an example XML file used by the TPS subsystem to configure the Phone Home feature.

```
<ServiceInfo><IssuerName>Example Corp</IssuerName>
    <Services>
        <Operation>http://tps.example.com:12443/nk_service ## TPS server URL
        </Operation>
        <UI>http://tps.example.com:12443/cgi_bin/esc.cgi   ## Optional
Enrollment UI
        </UI>
        <EnrolledTokenBrowserURL>http://www.test.url.com   ## Optional
enrolled token url
        </EnrolledTokenBrowserURL%gt;
    </Services>
</ServiceInfo>
```

Example 4.1. TPS Phone Home Configuration File

Phone Home is triggered automatically when a security token is inserted into a machine. The system immediately attempts to read the Phone Home URL from the token and to contact the TPS server.

If no Phone Home information is stored on the token, the the user is prompted for the Phone Home URL, as shown in *Figure 4.1, "Prompt for Phone Home Information"*. The other information is supplied and stored when the token is formatted. In this case, the company supplies the specific Phone Home URL for the user. After the user submits the URL, the format process adds the rest of the information to the Phone Home profile. The format process is not any different for the user.

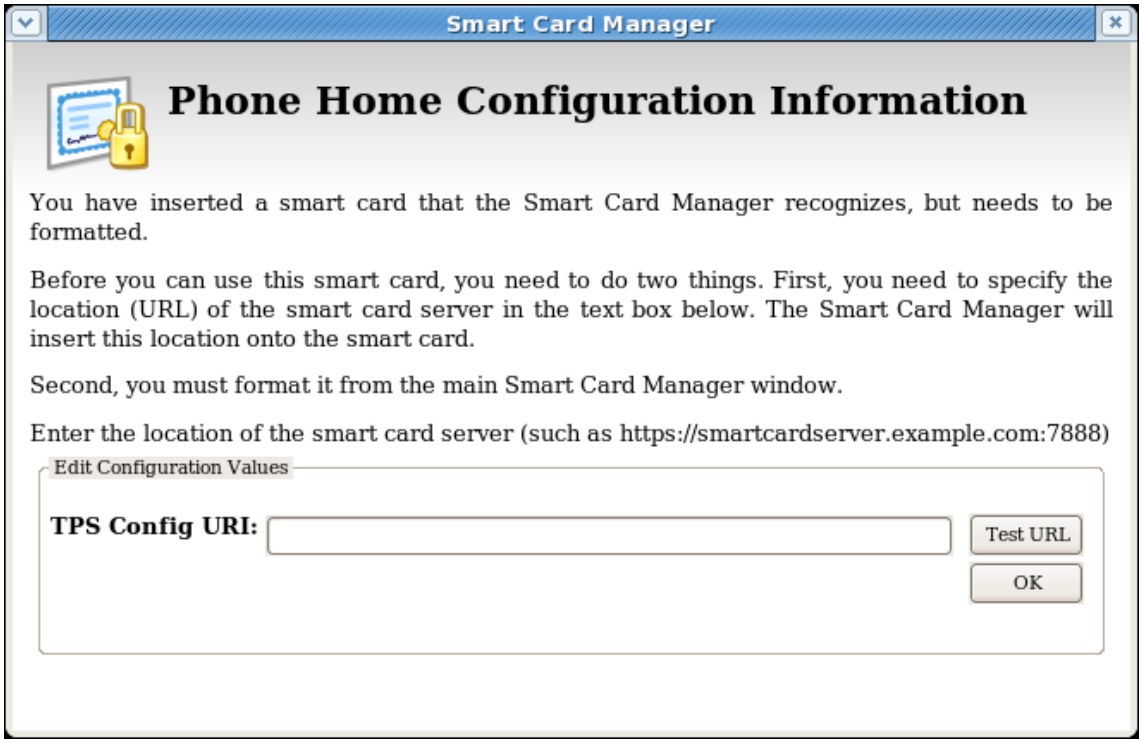

Smart Card Manager
Phone Home Configuration Information
You have inserted a smart card that the Smart Card Manager recognizes, but needs to be formatted.
Before you can use this smart card, you need to do two things. First, you need to specify the location (URL) of the smart card server in the text box below. The Smart Card Manager will insert this location onto the smart card.
Second, you must format it from the main Smart Card Manager window.
Enter the location of the smart card server (such as https://smartcardserver.example.com:7888)
Edit Configuration Values
TPS Config URI:
Test URL
OK

Figure 4.1. Prompt for Phone Home Information

The TPS configuration URI is the URL of the TPS server which returns the rest of the Phone Home information to the Enterprise Security Client. An example of this URL is `https://test.example.com:12443/cgi-bin/home/index.cgi`. When the TPS configuration URI is accessed, the TPS server is prompted to return all of the Phone Home information to the Enterprise Security Client.

The **Test** button can be used to test of the entered URL. If the server is successfully contacted, a message box indicates success. If the test connection fails, an error dialog appears.

3. Windows Cryptographic Service Provider

The Windows version of the Enterprise Security Client installs a *Windows Cryptographic Service Provider* (CSP) that is compatible with the Certificate System-supported smart cards.

Microsoft Windows supports a software library designed to implement the Microsoft Cryptographic Application Programming Interface (CAPI). CAPI allows Windows-based applications, such as the Windows-version of the Enterprise Security Client, to be developed to perform secure, cryptographic functions. This API, also known as CryptoAPI, provides a layer between an application which supports it, such as Certificate System, and the details of the cryptographic services provided by the API.

The CAPI interface can be used to create custom CSP libraries. In Certificate System, custom CSP libraries have been created to use the Certificate System-supported smart cards.

The *CAPI store* is a repository controlled by Windows that houses a collection of digital certificates associated with a given CSP. CAPI oversees the certificates, while each CSP controls the cryptographic keys belonging to the certificates.

The Certificate System CSP is designed to provide cryptographic functions on behalf of Windows using our supported smart cards. The Windows CSP performs its requested cryptographic functionality by calling the Certificate System PKCS #11 module.

The Certificate System CSP, which has been signed by Microsoft, provides the following features:

- Allows the user to send and receive encrypted and signed emails with Microsoft Outlook.
- Allows the user to visit SSL-protected websites with Microsoft Internet Explorer.
- Allows the user to use smart cards with certain VPN clients, which provides secure access to protected networks.

The required CSP libraries are automatically installed with the Enterprise Security Client. There are several common situations when a Windows user interacts directly with the CSP.

- When a smart card is enrolled with the Enterprise Security Client, the newly created certificates are automatically inserted into the user's CAPI store.
- When a smart card is formatted, the certificates associated with that card are removed from the CAPI store.
- When using applications like Outlook or Internet Explorer, the user may be prompted to enter the smart card's password. This is required when the smart card is asked to perform protected cryptographic operations such as creating digital signatures.

4. Smart Card Auto Enrollment

Because the Enterprise Security Client is configured through the Phone Home feature, simple enrollment of a smart card is extremely easy. Since the information needed to contact the backend TPS server is provided with each smart card, the enrollment process for the user is very simple.

Assuming that the smart card being enrolled is uninitialized and the appropriate Phone Home information has been configured, the user's enrollment process is as follows:

1. The Enterprise Security Client is running.
2. An uninitialized smart card, pre-formatted with the Phone Home information for the TPS and enrollment interface URL for the user's organization, is inserted. The smart card can be added either by placing a USB form factor smart card into a free USB slot or by inserting a

traditional full-sized smart card into a smart card reader.

3. When the system recognizes the smart card, it displays a message indicating it has detected an uninitiated smart card.

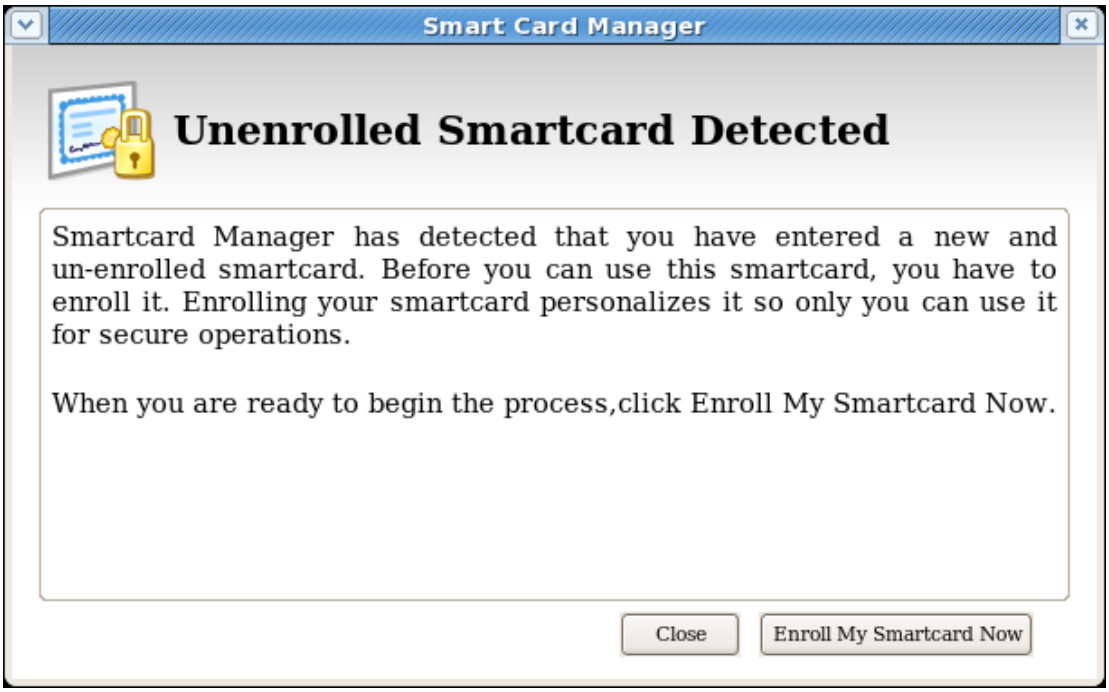

Smart Card Manager
Unenrolled Smartcard Detected
Smartcard Manager has detected that you have entered a new and un-enrolled smartcard. Before you can use this smartcard, you have to enroll it. Enrolling your smartcard personalizes it so only you can use it for secure operations.
When you are ready to begin the process,click Enroll My Smartcard Now.
Close
Enroll My Smartcard Now

Figure 4.2. Smart Card Enrollment with a Card

This screen gives the option either to close the dialog or to proceed with enrolling the smart card.

If the card be removed, a message appears that the smart card is no longer detected.

Smart Card Manager
No Smartcard Detected
Unable to detect your smartcard.
To enroll your smartcard, please insert your security card or USB key now.
Close

Figure 4.3. Smart Card Enrollment Message When the Card Is Removed

Reinserting the card brings the previous dialog back with the option to enroll the smart card. Click **Enroll My Smart Card** to continue with the enrollment process.

4. Since the Enterprise Security Client now knows where the enrollment UI is located because of Phone Home, the enrollment form opens for the user to enter the required information. This UI can be customized.

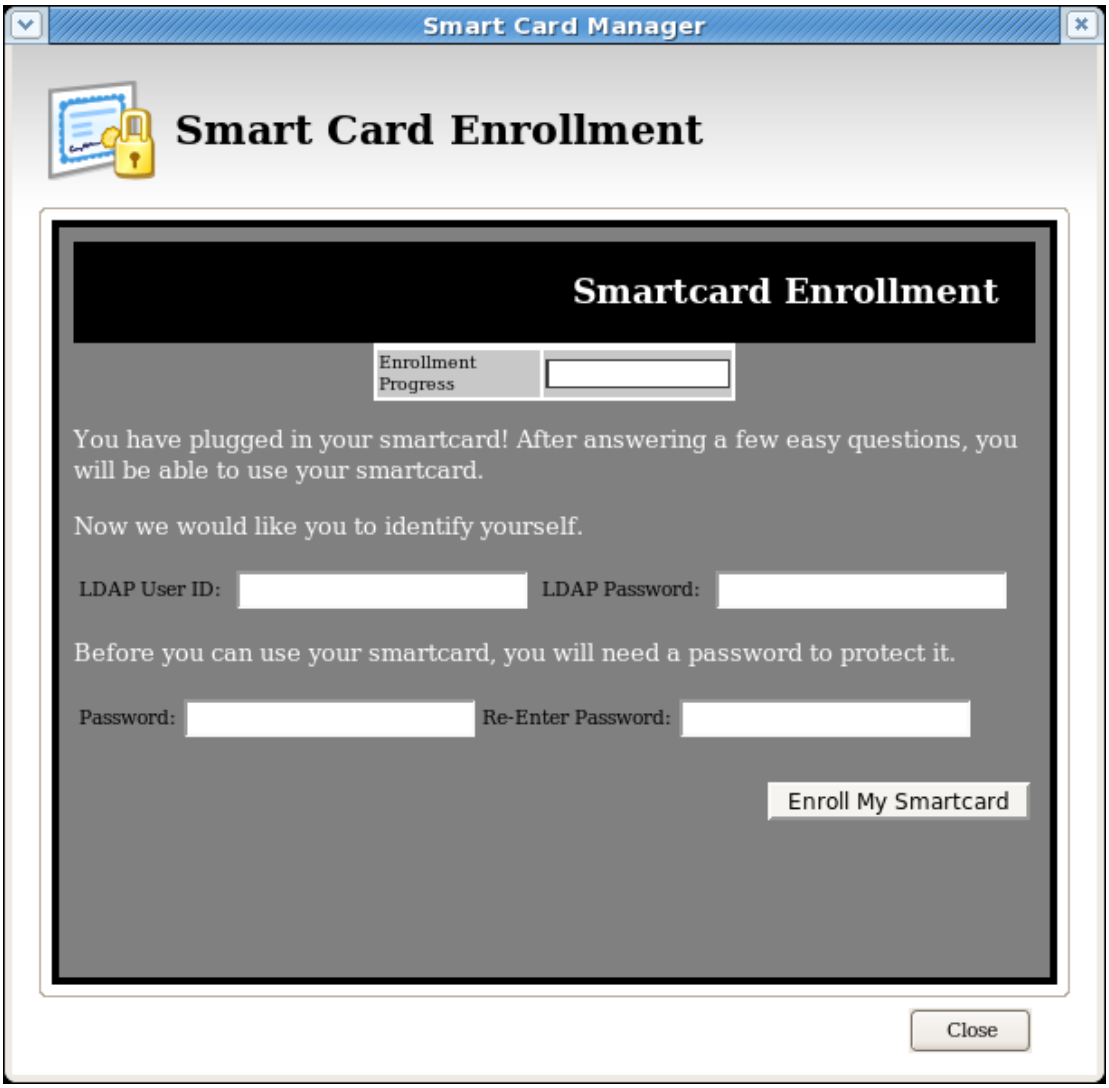

Smart Card Manager
Smart Card Enrollment
Smartcard Enrollment
Enrollment Progress
You have plugged in your smartcard! After answering a few easy questions, you will be able to use your smartcard.
Now we would like you to identify yourself.
LDAP User ID:
LDAP Password:
Before you can use your smartcard, you will need a password to protect it.
Password:
Re-Enter Password:
Enroll My Smartcard
Close

Figure 4.4. Smart Card Enrollment Page

5. This example is the default enrollment UI included with the TPS server. This UI is a standard HTML form, so simple modifications, such as setting the company logo or adding extra text or changing field text, is possible.

6. The sample enrollment UI requires the following information for the TPS server to process the smart card enrollment operation:

   - *LDAP User ID.* This is the LDAP directory user ID of the user enrolling the smart card; this can also be a screen name or employee or customer ID number.

   - *LDAP Password.* This is the alpha-numeric password corresponding to the user ID entered; this can be a simple password or a customer number.

NOTE

The LDAP user ID and password refer to the fact that the TPS server is usually associated with a Directory Server which stores user information and to which the TPS refers to authenticate users.

- *Password.* This sets the smart card's password, used to protect the card information.
- *Re-Enter Password.* This confirms the smart card's password.

7. Once the form is filled out, click **Enroll My Smartcard** to submit the information and enroll the card.
8. When the enrollment process is complete, a message page opens which shows that the card was successfully enrolled and can offer custom instructions on using the newly-enrolled smart card.

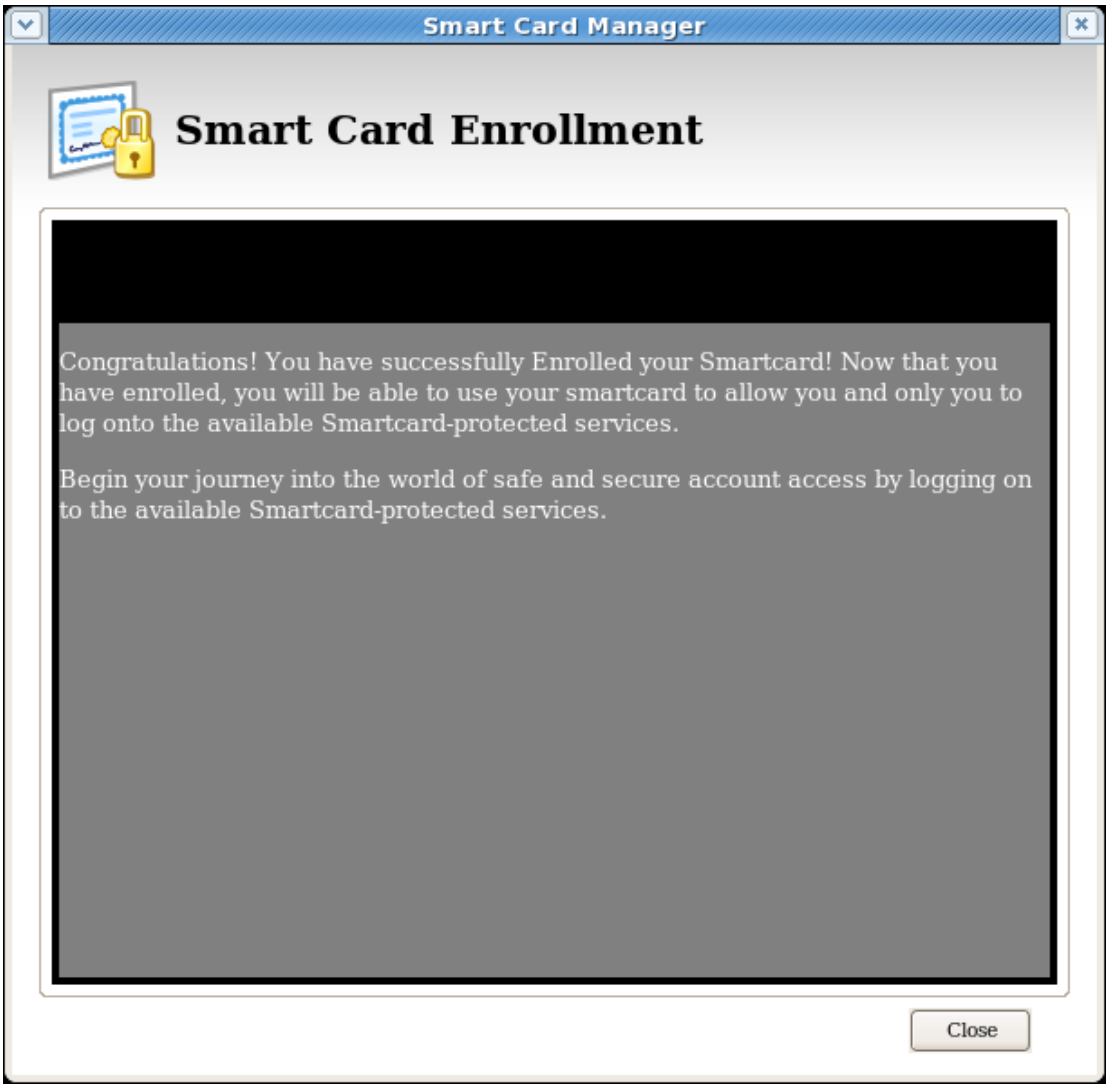

Smart Card Manager
Smart Card Enrollment
Congratulations! You have successfully Enrolled your Smartcard! Now that you have enrolled, you will be able to use your smartcard to allow you and only you to log onto the available Smartcard-protected services.
Begin your journey into the world of safe and secure account access by logging on to the available Smartcard-protected services.
Close

Figure 4.5. Smart Card Enrollment Success Message

5. Customizing the Smart Card Enrollment User Interface

The Certificate System TPS subsystem has a generic external smartcard enrollment user interface which is formatted in standard HTML and Javascript. This makes the interface page appearance easy to customize.

The default HTML file for the enrollment UI is located at `/var/lib/rhpki-tps/cgi-bin/home/Enroll.html` The UI references resources such as images and Javascript files within its code. These resources are located in `/var/lib/rhpki-tps/docroot/home/`

The default HTML page, shown in *Example 4.2, "Customizing the Smart Card Enrollment User Interface"*, can be edited to change the colors, images, and layout.

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <link rel=stylesheet href="/home/style.css" type="text/css">

          <!-- Change the title if desired -->
          <title>Enrollment</title>

   </head>

   <script type="text/JavaScript" src="/home/util.js">
   </script>
   <body onload="InitializeBindingTable();" onunload=cleanup();>
          <progressmeter id="progress-id" hidden="true" align = "center"/>

          <table width="100%" class="logobar">
             <tr>
                    <td>

                       <!-- Use customized logo here... -->
                       <img alt="" src="/home/logo.jpg">

                    </td>
                    <td>
                       <p class="headerText">Smartcard Enrollment</p>
                    </td>
             </tr>
          </table>

          <table id="BindingTable" width="200px"align="center">
             <tr id="HeaderRow">
             </tr>
          </table>

          <!-- Insert customized descriptive text here. -->
          <p class="bodyText">You have plugged in your smartcard!
             After answering a few easy questions, you will be able to use
your smartcard.
          </p>

          <p class="bodyText">
             Now we would like you to identify yourself.
          </p>

          <table>
             <tr>
                    <td><p >LDAP User ID: </p></td>
                    <td> </td>
                    <td><input type="text" id="snametf"  value=""></td>
                    <td> </td>
                    <td><p>LDAP Password: </p></td>
                    <td> </td>
```

```
                    <td><input type="password" id="snamepwd" value=""></td>
               </tr>
            </table>

            <p class="bodyText"> Before you can use your smartcard, you need a
password to protect it.</p>

            <table>
               <tr>
                    <td><p >Password:</p></td>
                    <td><input type="password" id="pintf" name="pintf"
value=""></td>
                    <td><p >Re-Enter Password:</p></td>
                    <td><input type="password" id="reenterpintf"
name="reenterpintf" value=""></td>
               </tr>
            </table>
            <br>
            <table width="100%">
               <tr>
                    <td align="right">
                    <input type="button" id="enrollbtn" name="enrollbtn"
value="Enroll My Smartcard"
                         onClick="DoEnrollCOOLKey();">
                    </td>
               </tr>
            </table>
      </body>
</html>
```

Example 4.2. Customizing the Smart Card Enrollment User Interface

6. Managing Smart Cards

The **Manage Smart Cards** page contains many of the operations that can be applied to one of the keys. This page allows users to format the token, set and reset the card's password, and show the card information. Two other operations, enrolling tokens and viewing the diagnostic logging, are also accessed through the **Manage Smart Cards** page. These are addressed in other sections.

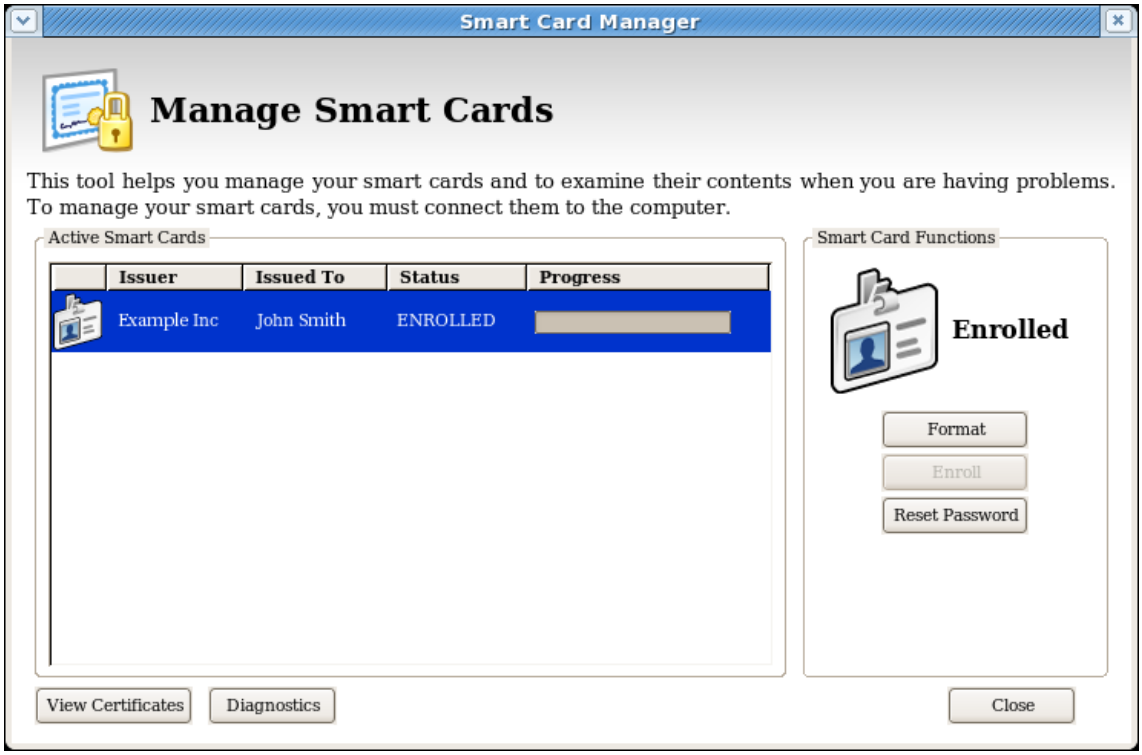


Figure 4.6. Manage Smart Cards Page

6.1. Formatting the Smart Card

Formatting the card brings the smart card to the uninitialized state, which removes all the user keypairs previously generated and erases the password set on the smart card during enrollment.

The TPS server can be configured to load newer versions of the applet and symmetric keys onto the card. Do the following to format the smart card:

1. Place a supported smart card into the USB slot of the computer. Make sure the card shows up in the **Active Smart Cards** table.
2. Select **Format** from the **Smart Card Functions** section in the **Manage Smart Cards** screen.
3. The TPS can be configured to authenticate smart card operations using credentials such as an LDAP user ID and password. If the TPS has been configured for user authentication, fill in the user credentials in the authentication prompt, and click **OK**.
4. Wait for the token to finish being formatted. A success message will be displayed when the formatting operation is complete.
5. When the formatting is complete, the **Active Smart Cards** table shows the key as uninitialized.

6.2. Reset Smart Card Password

If a user forgets the password for a smart card after the card is enrolled, it is possible to reset the password by doing the following:

1. Place a supported smart card into the USB slot of the computer. Make sure the smart card shows up in the **Active Smart Cards** table.
2. Click the **Reset Password** button in the **Smart Card Functions** window. A dialog for resetting the password on the card then opens.
3. Enter a new smart card password value in the **Enter new password** field.
4. Confirm the new smart card password value in the **Re-Enter password** field.

Figure 4.7. Changing Password Dialog

5. The TPS can be configured to authenticate smart card operations using credentials such as an LDAP user ID and password. If the TPS has been configured for user authentication, fill in the user credentials in the authentication prompt.
6. Wait for the token password to finish being reset.

6.3. Viewing Certificates

The **View Certificates** button shows basic information about the selected smart card, including the keys and certificates stored on it.

1. Place a supported smart card into the USB slot of the computer. Make sure the card shows up in the **Active Smart Cards** table.
2. Select the card from the list.
3. Press the **View Certificates** button.
4. Basic information about the certificates stored on the card is shown, including the serial number, certificate nickname, and validity dates. More detailed information about the certificate can be viewed by selecting a certificate from the list, and clicking **View**.

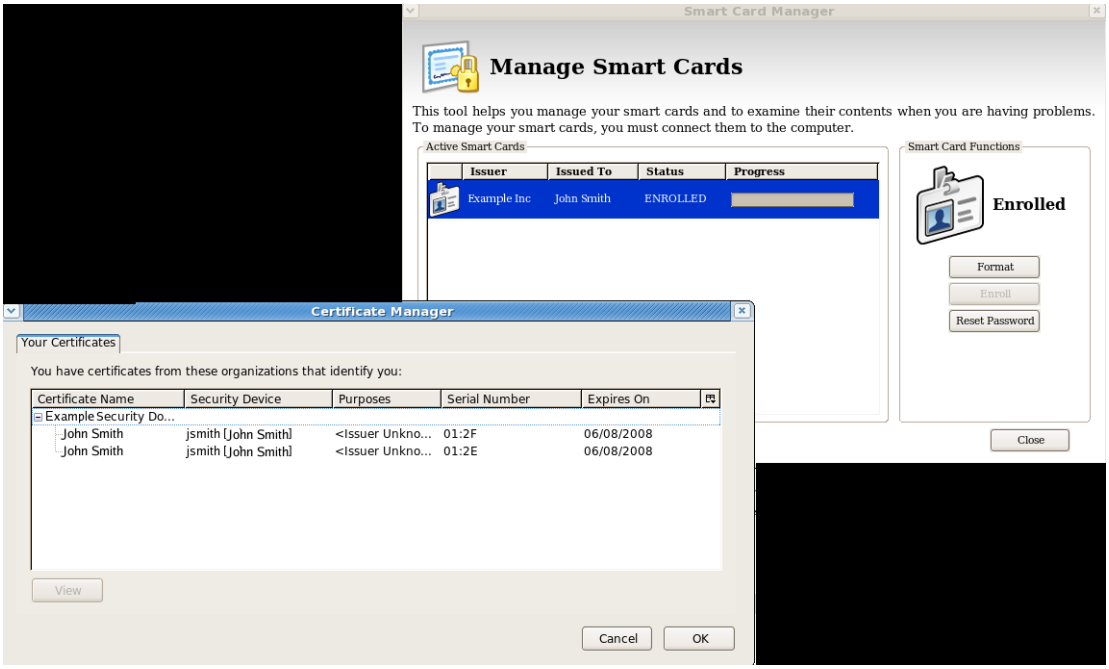


Figure 4.8. Viewing Certificates

6.4. Enrolling Smart Cards

Although most smart cards will be enrolled using the automated enrollment, described in *Section 4, "Smart Card Auto Enrollment"*, there is an alternative way of enrolling smart cards through the **Manage Smart Cards** page.

Red Hat Enterprise Security Client
Settings
Configuration
Red Hat Enterprise Security Client
Please Enroll Your Key
Please Enroll your key. Enrolling your key involves setting up a password. After you set your password, you'll need to enter it every time you would like to use your key. This adds additional security to the use of your key.
Detected Key Details
New Key
Issuer
Status
Issued
Unknown Issuer
UNINITIALIZED
Unknown
KeyEnrollment
Please enter a password for your key:
Password Quality
GOOD
Please re-enter above password
Close,I'll Enroll Later
Enroll Key

Figure 4.9. Manual Enrollment Form

Enrolling a token with the user key pairs means the token can be used for certificate-based operations such as SSL client authentication and S/MIME.

NOTE

The TPS server can be configured to generate the user key pairs on the server and then archived in the DRM subsystem for recovery if the token is lost.

1. Place a supported, unenrolled, smart card into the USB slot of the computer. Make sure the card shows up in the **Active Smart Cards** table at the top.
2. Press the **Enroll** button. This button is active only if the inserted card is unenrolled.
3. A dialog opens which is used to set the password on the smart card. Enter a new key password value in the **Enter a password** field.

   Confirm the new card password value in the **Re-Enter a password** field.
4. Click **OK** to begin the enrollment.

5. The TPS server can be configured to authenticate the enrollment operation. If the TPS has been configured for authentication, enter the user credentials when the dialog box appears, and click **OK**.

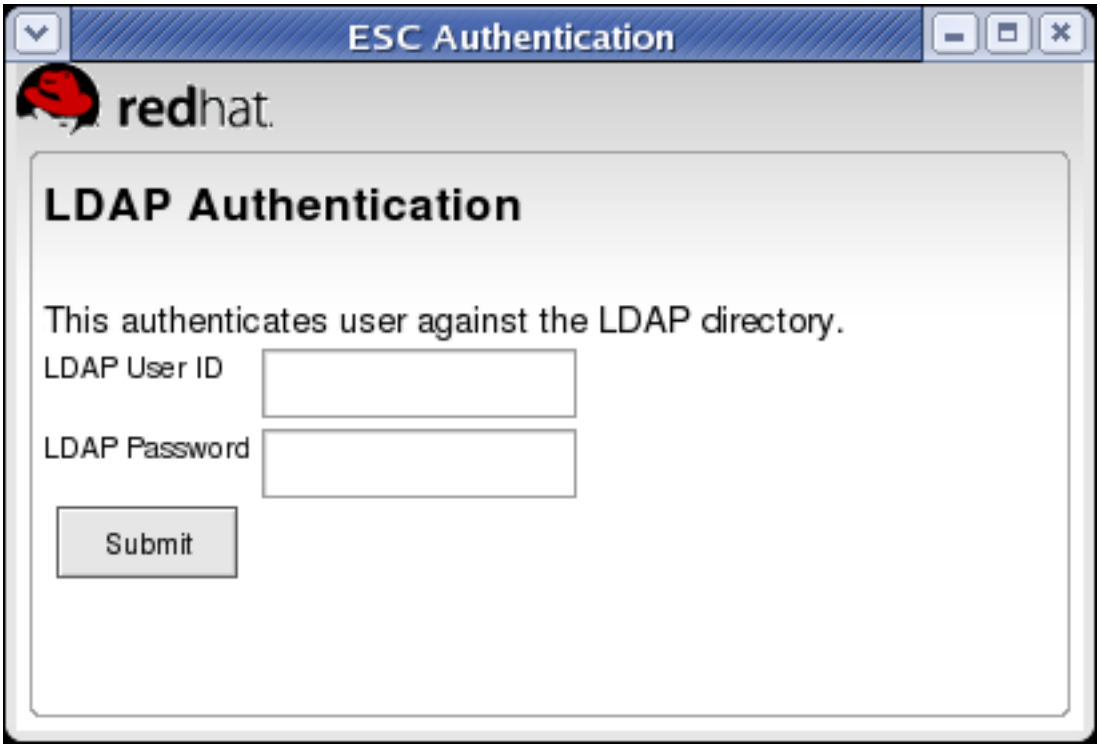


Figure 4.10. LDAP Authentication Prompt

6. The enrollment process will begin generating and archiving keys, if the TPS is configured to archive keys to the DRM.

7. When the enrollment is complete, the smart card will be listed as `enrolled`.

7. Diagnosing Problems

The Enterprise Security Client includes basic diagnostic tools and a simple interface to log errors and common events such as inserting and removing a smart card or changing the card's password. The diagnostic tools can identify and notify users about problems with the Enterprise Security Client, smart cards, and TPS connections.

To open the diagnostics page, click on the **Diagnostics** button in the **Manage Smart Cards** screen.

The following problems or events are logged by the Enterprise Security Client:

- The Enterprise Security Client does not recognize a card.

- Problems occur during a smart card operation, such as a certificate enrollment, password reset, or format operation.
- The Enterprise Security Client loses the connection to the smart card. This can happen when problems communicating with the PCSC daemon.
- Simple events are detected, such as when a card is inserted or removed, a user cancels an operation, an operation is successfully completed, or errors are reported from the TPS.
- The connection between the Enterprise Security Client and TPS is lost.
- The NSS crypto library is initialized.
- Other low-level smart card events are detected.

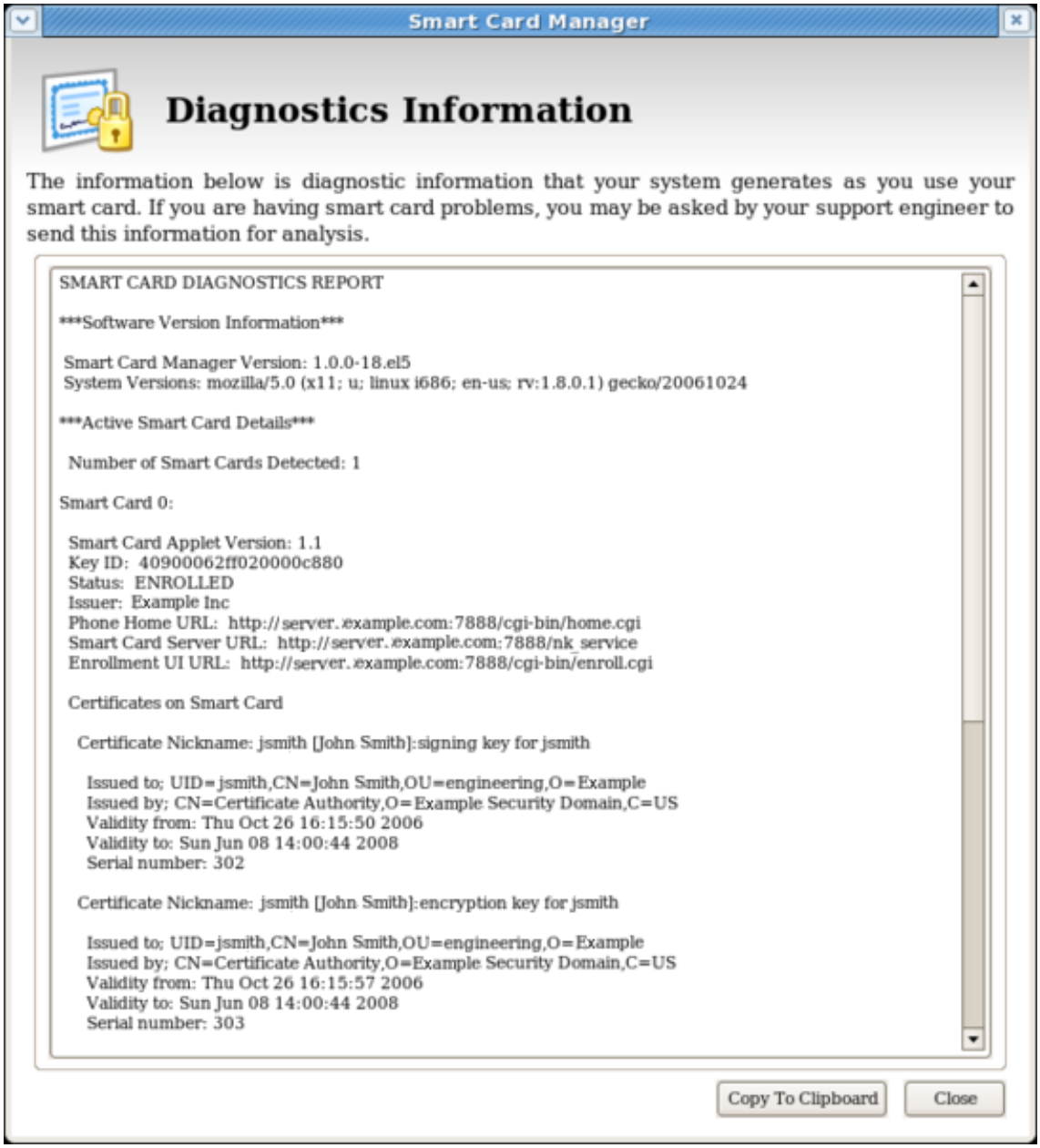

Smart Card Manager
Diagnostics Information
The information below is diagnostic information that your system generates as you use your smart card. If you are having smart card problems, you may be asked by your support engineer to send this information for analysis.
SMART CARD DIAGNOSTICS REPORT
Software Version Information
Smart Card Manager Version: 1.0.0-18.el5
System Versions: mozilla/5.0 (x11; u; linux i686; en-us; rv:1.8.0.1) gecko/20061024
Active Smart Card Details
Number of Smart Cards Detected: 1
Smart Card 0:
Smart Card Applet Version: 1.1
Key ID: 40900062ff020000c880
Status: ENROLLED
Issuer: Example Inc
Phone Home URL: http://server.example.com:7888/cgi-bin/home.cgi
Smart Card Server URL: http://server.example.com:7888/nk_service
Enrollment UI URL: http://server.example.com:7888/cgi-bin/enroll.cgi
Certificates on Smart Card
Certificate Nickname: jsmith [John Smith]:signing key for jsmith
Issued to; UID=jsmith,CN=John Smith,OU=engineering,O=Example
Issued by; CN=Certificate Authority,O=Example Security Domain,C=US
Validity from: Thu Oct 26 16:15:50 2006
Validity to: Sun Jun 08 14:00:44 2008
Serial number: 302
Certificate Nickname: jsmith [John Smith]:encryption key for jsmith
Issued to; UID=jsmith,CN=John Smith,OU=engineering,O=Example
Issued by; CN=Certificate Authority,O=Example Security Domain,C=US
Validity from: Thu Oct 26 16:15:57 2006
Validity to: Sun Jun 08 14:00:44 2008
Serial number: 303
Copy To Clipboard
Close

Figure 4.11. Diagnostics Screen

The diagnostics screen displays the following information:

- The Enterprise Security Client version number.
- The version information for the system upon which the client is running.
- The number of cards detected by the Enterprise Security Client.

For each card detected, the following information is shown:

- The version of the applet running inside the smart card.
- The alpha-numeric ID of the card.
- The card's status, which can be `NO_APPLET` (no key is detected), `UNINITIALIZED` (the key is detected, but no certificates have been enrolled), or `ENROLLED` (the detected card has been enrolled with certificate and card information).
- The card's Phone Home URL. This is the URL from which all Phone Home information is obtained.
- The card issuer name, such as `Example Corp.`
- The TPS server URL. This is retrieved through Phone Home.
- The TPS enrollment form URL. This is retrieved through Phone Home.
- Detailed information about each certificate contained on the card.

Using Enterprise Security Client Keys for SSL Client Authentication and S/MIME

After a token is enrolled, the token can be used for SSL client authentication and S/MIME email applications.

The PKCS #11 module has different names and locations depending on the operating system.

Platform	Module Name	Location
Windows	coolkeypk11.dll	C:\Windows\System32\
Red Hat Enterprise Linux	libcoolkeypk11.so	/usr/lib/
Mac	libcoolkeypk11.dylib	/Library/Application Support/CoolKey/PKCS11

1. Using the Certificates on the Token for SSL

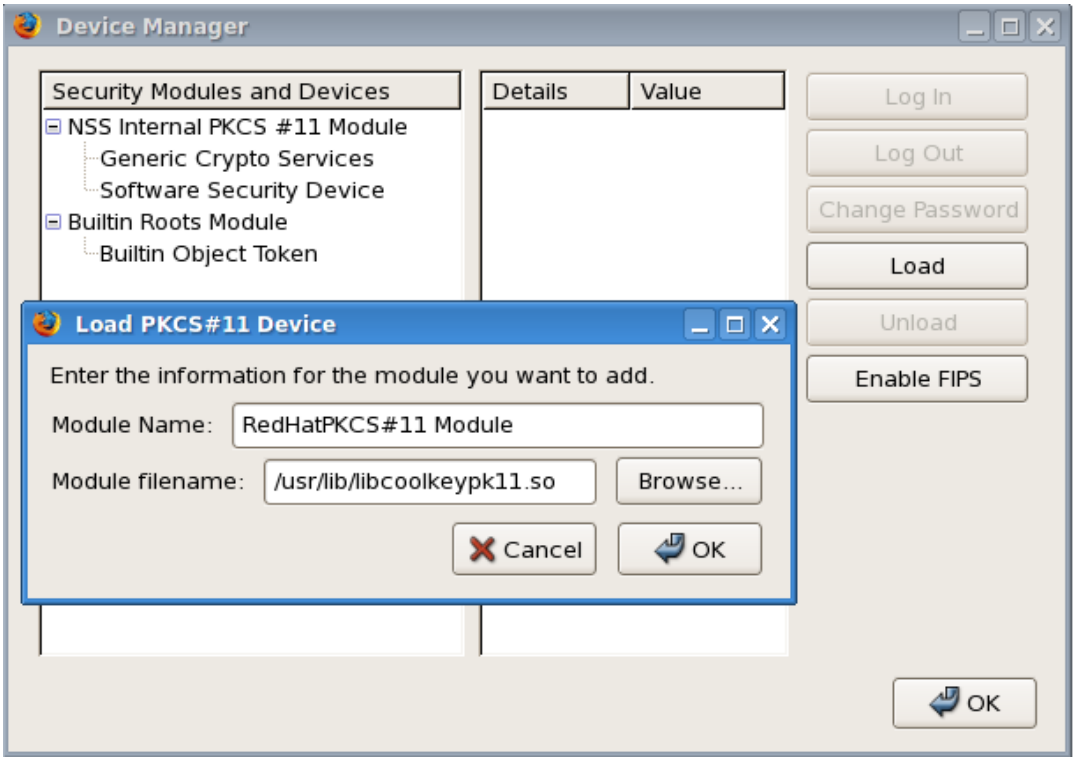
To use the certificate on the token for SSL in an application such as Mozilla Firefox:

1. In Mozilla Firefox, open the **Tools** menu, choose **Options**, and then click **Advanced**.
2. Add a PKCS #11 driver.

NOTE

Windows and Macs automatically attempt to load the PKCS #11 module to any Mozilla browsers they find.

   a. Click **Manage Security Devices** to open the **Device Manager** window, and then click the **Load** button.
   b. Enter a module name, such as `token key pk11 driver`.
   c. Click **Browse**, find the Enterprise Security Client PKCS #11 driver, and click **OK**.



3. If the CA is not yet trusted, download and import the CA certificate.

   a. Open the **SSL End Entity** page on the CA. For example:

```
https://example.com:9443/ca/ee/ca
```

   b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.

   c. Click **Download the CA certificate chain in binary form** and then click **Submit**.

   d. Choose a suitable directory to save the certificate chain, and then click **OK**.

   e. Click **Edit > Preferences**, and select the **Advanced** tab.

   f. Click the **View Certificates** button.

   g. Click **Authorities**, and import the CA certificate.

4. Set the certificate trust relationships.

   a. Click **Edit > Preferences**, and select the **Advanced** tab.

   b. Click the **View Certificates** button.

   c. Click **Edit**, and set the trust for websites.

The certificates can be used for SSL.

2. S/MIME Applications

To enable S/MIME on mail applications such as Mozilla Thunderbird:

1. In Mozilla Thunderbird, open the **Edit** menu, and select **Account Settings**.
2. Select **Security** on the left.
3. Add a PKCS #11 driver.
   a. Click **Manage Security Devices** to open the **Device Manager** window.
   b. Click the **Load** button.
   c. Enter the module name, such as `token keypk11 driver`.
   d. Click **Browse**, find the Enterprise Security Client PKCS #11 driver, and click **OK**.
4. If the CA is not yet trusted, download and import the CA certificate.
   a. Open the **SSL End Entity** page on the CA. For example:

```
https://example.com:9443/ca/ee/ca
```

   b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.
   c. Click **Download the CA certificate chain in binary form** and then click **Submit**.
   d. Choose a suitable directory to save the certificate chain, and then click **OK**.
   e. In Thunderbird, open the **Edit** menu, and select **Account Settings**.
   f. Select **Security** on the left, and click the **Manage Certificates** button.
   g. Click the **Authorities** tab, and import the CA certificate.
5. Set up the certificate trust relationships.
   a. In Thunderbird, open the **Edit** menu, and select **Account Settings**.
   b. Select **Security** on the left, and click the **Manage Certificates** button.
   c. In the **Authorities** tab, select the CA, and click the **Edit** button.
   d. Set the trust settings for identifying websites and mail users.
   e. In the **Digital Signing** section of the **Security** panel, click **Select** to choose a certificate to use for signing messages.

6. In the **Encryption** of the **Security** panel, click **Select** to choose the certificate to encrypt and decrypt messages.

Uninstalling Enterprise Security Client

This section provides platform-specific instructions to uninstall Enterprise Security Client.

1. Uninstalling on Windows

1. Unplug all USB tokens.
2. Stop Enterprise Security Client.
3. Open the **Control Panel**, and select the **Add Remove Programs** icon.
4. Select **Smart Card Manager** from the list of programs, and and click **Change/Remove**.
5. When the uninstallation is complete, remove any files left in the installation directory.

2. Uninstalling on Red Hat Enterprise Linux

1. Unplug all USB tokens.
2. Stop Enterprise Security Client.
3. Log in as `root`, and use `rpm -ev` to remove the Enterprise Security Client RPMs in the following order:

```
rpm -ev ccid-1.0.1-5.i386.rpm
rpm -ev pcsc-lite-1.3.1-7.i386.rpm
rpm -ev pcsc-lite-libs-1.3.1-7.i386.rpm
rpm -ev ifd-egate-0.05-15.i386.rpm
rmp -ev coolkey-1.0.1-4.i386.rpm
rpm -ev esc-1.0.0-16.i386.rpm
```

4. Remove any files left in the installation directory.

3. Uninstalling on Mac OS X

1. Unplug all USB tokens.
2. Stop Enterprise Security Client.
3. Send the `ESC.app` icon to the trash.

NOTE

There is no uninstallation program for the Mac.

Appendix A. Enterprise Security Client Configuration

Previously, Enterprise Security Client relied on an application-specific configuration file. Enterprise Security Client is now based on Mozilla XULRunner technology, which allows the preferences facility built into Mozilla to be used for simple configuration of the Enterprise Security Client. A simple UI, discussed in *Chapter 4, Using the Enterprise Security Client*, manages most important configuration settings.

NOTE

The Enterprise Security Client can be launched without requiring extra configuration.

1. Configuration

Enterprise Security Client uses the Mozilla configuration preferences system on all three supported platforms. A default configuration file is located in the following directories on each platform:

- *Windows*: `C:\Program Files\Red Hat\ESC\defaults\preferences\esc-prefs.js`
- *Red Hat Enterprise Linux*: `/usr/lib/esc-1.0.0/esc/defaults/preferences/esc-prefs.js`
- *Mac*: `~/Desktop/ESC.app/defaults/preferences/esc-prefs.js`

This default configuration Javascript file contains the default Enterprise Security Client configuration used when Enterprise Security Client is first launched.

When Enterprise Security Client is launched, a separate, unique profile directory for each user on the system is created; these profiles are stored in different, accessible locations on each platform, as shown:

Note

When the Enterprise Security Client requires any changes to a user's configuration values, the updated values are written to the user's profile area, not to the default Javascript file.

- *Windows*: `C:\Documents and Settings\$USER\Application Data\RedHat\ESC\Profiles`
- *Red Hat Enterprise Linux*: `~/.redhat/esc`
- *Mac*: `~/Library/Application Support/ESC/Profiles`

The `esc-prefs.js` file section below shows the Enterprise Security Client-supported configuration values.

```
######################################################################
#The entry below is the XUL chrome page where Enterprise Security
#Client proceeds on startup.
#
 pref("toolkit.defaultChromeURI",
  "chrome://esc/content/settings.xul");

#The entry below is the URL Enterprise Security Client consults
#for back end TPS functionality.
 pref("esc.tps.url","https://test.host.com:7888/nk_service");

#The following three entries are for internal use
 pref("signed.applets.codebase_principal_support",true);
 pref("capability.principal.codebase.p0.granted",
    "UniversalXPConnect");
 pref("capability.principal.codebase.p0.id", "file://");

#The entry below sets how many seconds Enterprise Security Client
#should wait while TPS is processing a message
 pref("esc.tps.message.timeout","90");

#The entry can be set allow Enterprise Security Client to write
#newly created certificates
#to the local CAPI store after an enrollment operation.
#Also, when a format is done, those same certs will be removed
#from the local CAPI store.

pref("esc.windows.do.capi","yes");
######################################################################
```

Example A.1. Example Configuration File

2. Enterprise Security Client Mac TokenD

The TokenD software installed on Mac provides a link between the Certificate System CoolKeys and the Mac CDSA security API, which provides a wide variety of security functionality. For example, the Apple Mail application can use a *KeyChain* to perform security-related tasks. A KeyChain can hold entities such as certificates, passwords, and private and public keys. Although most KeyChains are stored in software, the CDSA API allows KeyChains to be stored

on smart cards or keys. CoolKey TokenD allows a Certificate System key to show as a KeyChain.

2.1. Verifying the TokenD Is Working

1. Make sure Enterprise Security Client has been installed on the Mac computer.
2. Use Enterprise Security Client to enroll a token, enabling it with the proper certificates and key information.
3. Put the enrolled token into a USB slot on the machine.
4. If TokenD is working, the token blinks for a few seconds while the information is obtained from the token because the Mac CDSA layer is making a request for data.
5. Open the Mac `Keychain Access` utility in the `Applications/Utilities` folder.
6. Find the new `$Keychain` entry in the list of valid chains. The chain has the the key's UID in its name.
7. Click on the CoolKey KeyChain. The certificates and keys on the token can easily be viewed.

3. Enterprise Security Client XUL and Javascript Functionality

Enterprise Security Client stores the XUL markup and Javascript functionality in the *ESC_INSTALL_PATH*`/chrome/content/esc` directory, where *ESC_INSTALL_PATH* is the Enterprise Security Client installation directory.

The following are the primary Enterprise Security Client XUL files:

- `settings.xul` contains the code for the **Settings** page.
- `esc.xul` contains the code for the **Enrollment** page.
- `config.xul` contains the code for the configuration UI.
- `esc_browser.xul` contains the code for hosting the external HTML Enterprise Security Client enrollment UI.

The following are the primary Enterprise Security Client Javascript files:

- `ESC.js` contains most of the Enterprise Security Client Javascript functionality.
- `TRAY.js` contains the tray icon functionality.
- `CertInfo.js` contains the code for **Show Key Info** feature.

- `GenericAuth.js` contains the code for the authentication prompt. This prompt is configurable from the TPS server, which requires dynamic processing by Enterprise Security Client.

3.1. Quick Javascript UI Guide

Certificate System 7.1 deployments may be using a customized external UI for key enrollment. Changes have been made to the names of internal Enterprise Security Client XPCOM objects in later versions of Certificate System, so changes need to be made to the `ESC.js` file to adapt an older UI. The places for these changes are shown in the file section below.

```
//ESC.js : Core Enterprise Security Client functionality
....
//
// Attach to the Enterprise Security Client XPCOM object on load
//
  try {
netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect");
    netkey = Components.classes["@redhat.com/rhCoolKey"].getService();
    netkey = netkey.QueryInterface(Components.interfaces.rhICoolKey);
    gNotify = new jsNotify;
    netkey.rhCoolKeySetNotifyCallback(gNotify);
  } catch(e) {
     alert("Can't get UniversalXPConnect: " + e);
  }

//Sample function to complete Enrollment of a key.

function EnrollCoolKey(keyType, keyID, enrollmentType, screenname,
                       pin,screennamepwd,tokencode)
 {
   try {

     netkey.EnrollCoolKey(keyType, keyID, enrollmentType, screenname,
                          pin,screennamepwd,tokencode);
   } catch(e) {
     ReportException("netkey.EnrollCoolKey() failed!", e);
     return false;
   }

   return true;
 }
```

4. Enterprise Security Client File Locations

This reference shows the different directories and file locations for the different client machines.

- The location of the Enterprise Security Client main directory on the different client platforms is as follows:

- *Windows*: `C:\Program Files\Red Hat\ESC`
- *Red Hat Enterprise Linux*: `/usr/lib/esc-1.0.0/esc`
- *Mac*: User preference for the `ESC.app` directory, usually the desktop

4.1. Windows

On Windows, Enterprise Security Client uses the following directories and files:

- Main directory: `C:\Program Files\Red Hat\ESC`
  - Enterprise Security Client XULRunner application configuration file: `application.ini`
  - Enterprise Security Client XPCOM components directory: `components\`
  - Directory for Chrome components and additional application files for Enterprise Security Client XUL and Javascript: `chrome\`
  - Enterprise Security Client default preferences: `defaults\`
  - The executable which launches Enterprise Security Client in XULRunner: `esc.exe`
  - Privately-deployed XULRunner bundle: `xulrunner\`

4.2. Red Hat Enterprise Linux

On Linux, Enterprise Security Client is installed by its binary RPM to the default location `/usr/lib/esc-1.0.0/esc/`.

- Enterprise Security Client XULRunner application configuration file: `application.ini`
- Enterprise Security Client XPCOM components: `components/`
- Directory for Chrome components and additional application files for Enterprise Security Client XUL and Javascript: `chrome/`
- Enterprise Security Client default preferences: `defaults/`
- The script which launches the Enterprise Security Client: `esc`
- Privately-deployed XULRunner directory: `xulrunner/`

4.3. Mac OS X

On Mac OS X, the XULRunner framework located in `ESC.app` as follows:

- Privately deployed XUL framework in `Contents/`
  - `Info.plist`
  - `Frameworks/`
  - `XUL.framework/`
  - Resources
    - Enterprise Security Client XULRunner application configuration file: `application.ini`
    - Enterprise Security Client XPCOM components: `components/`
    - Directory for Chrome components and additional application files for Enterprise Security Client XUL and Javascript: `chrome/`
    - Enterprise Security Client default preferences: `defaults/`
- The script which launches Enterprise Security Client: `xulrunner`

Index